# Using the Standard Libraries

# 21

This chapter describes what standard libraries come with THINK C, how to use them with your project, and how to modify them for your own purposes. It also explains how to use the online reference to look up functions.

## Contents


What Are the Standard Function Libraries? . . . . . 401
- Special versions of the standard libraries . . . . . 402
- Customizing the ANSI library . . . . . 403

Using the Online Standard Libraries Reference . . . . . 404
- Looking up a function . . . . . 405
- Looking up a topic . . . . . 406
- Moving around THINK Reference . . . . . 406
- Reading a function reference page . . . . . 407
- Using the tables of contents . . . . . 408
- Finding lost databases . . . . . 410

## What Are the Standard Function Libraries?

*The Profile library is also in the* `Standard Libraries` *folder. It's described in Chapter 25, "Profiling Projects."*

THINK C includes two standard function libraries, ANSI and UNIX, which you can find in the `Standard Libraries` folder:

- The ANSI library is a complete implementation of the ANSI standard library. The standard gives all C programmers in all ANSI C environments a consistent set of functions, making programs easier to port. It includes functions to perform file and screen I/O, string-handling, math, and more.

- The UNIX library contains several functions common to Unix implementations of C that are not part of the ANSI standard library. It includes functions to handle files, signals, and other Unix-specific features.

Use the UNIX library to make porting software from Unix systems easier. When writing a new program, try to avoid the UNIX library, and use the ANSI library instead. The descriptions of the Unix functions in the online *Standard Libraries Reference* tell you when there is an equivalent function in the ANSI library.

The ANSI library includes the console package, which lets you use simple Macintosh windows called consoles. Consoles let you port MS-DOS and Unix programs easily and help you get simple C programs running quickly. For more information, see "Using the Online Standard Libraries Reference" later in this chapter.

Symantec C++ includes five standard function libraries, ANSI++, UNIX++, CPlusLib, IOstreams, and Complex, which you can find in the `Standard Libraries` folder:

- The ANSI++ library (see above).

- The UNIX++ library (see above).

- The CPlusLib library contains utility routines for the Symantec C++ compiler that handle the new and delete operators for objects and arrays. It also contains routines that handle certain run-time math operations.

- The IOstreams library is a full implementation of the standard stream input/output library described in Bjarne

Stroustrup's *The C++ Programming Language*. It provides flexible and extendable facilities for input and output.

- The Complex library allows you to perform arithmetic operations and complex numbers.

### Special versions of the standard libraries

The special versions of the standard libraries are designed to handle special circumstances. Note that these libraries leave out some functions; therefore, use one of these libraries only if your program meets the condition for which that library was designed:

| For THINK C use this library… | If your program… |
|---|---|
| ANSI-small | Does not use consoles or floating point functions |
| ANSI-A4 | Is a desk accessory, device driver, or other kind of code resource |

| For Symantec C++ use this library… | If your program… |
|---|---|
| ANSI-small++ | Does not use consoles or floating-point functions |
| ANSI-A4++ | Is a desk accessory, device driver, or other kind of code resource |
| CPlusLib-A4 | Is a desk accessory, device driver, or other kind of code resource |

**Table 21-1** Special versions of the standard libraries

The special ANSI libraries differ from the ANSI library as follows:

- The ANSI-small library is smaller than ANSI, and projects that use it are smaller as well. It leaves out many things that are available in the ANSI library. Most importantly, it leaves out console functions and floating-point support. Also, the `%f`, `%g`, and `%e` conversions are not available for the formatted I/O functions, such as `printf()` and `scanf()`.

- The ANSI-A4 library is a subset of ANSI-small. It leaves out everything that you can't use outside of applications. In addition to all the things ANSI-small leaves out, it leaves out file I/O functions. However, it includes string-handling functions, such as `sscanf()` and `sprintf()`. This library is for device drivers, desk accessories, and code resources only. The other standard libraries should be used in applications only.

### Customizing the ANSI library

You may want to create your own version of the ANSI library. For example, you may want one for Macintosh computers with an MC68881 math coprocessor or one that includes floating-point functions and leaves out console functions. Here's how to do it:

1. Choose the library from which you want to start. For example, if you want your library to exclude console and floating-point functions, start with ANSI-small.

2. In the Finder, create a copy of the project and give it a new name, like `ANSI-881` or `MyANSI`.

3. In the THINK Project Manager, open your renamed project.

4. In the **THINK C options** dialog, set up your options. Some common settings are listed below.

5. Remove files you don't need. For example, if you don't need time functions, remove `time.c`.

6. Choose **Bring Up To Date** from the **Project** menu to compile your project.

---

Note

If you're creating a library to use with Symantec C++, remember to turn on the options 4-byte ints and Native floating-point format.

---

If you remove files from the ANSI library, you may get link errors when you compile your project. The Link Errors window lists the functions that other functions in your project need. Search the files you removed, and add back to your project the files that contain the needed functions to your project.

Here are some common options settings:

| To create a library that uses… | Set these options… |
|---|---|
| No console functions | In the Prefix section, define the symbol `_NOCONSOLE_`, and remove the files `console.c` and `command.c`. |
| No floating-point functions | In the Prefix section, define the symbol `_NOFLOATING_`. |
| The MC68881, directly | Turn on the Generate 68881 instructions option. |
| SANE floating-point format | Turn off the Generate 68881 instructions option, and turn on the Native floating-point format option. |
| MC68881 floating-point format | Turn on the Generate 68881 instructions option, and turn on the Native floating-point format option. |

**Table 21-2** Common option settings for the ANSI library

You can combine some of these options. For example, you can make a library for Macintosh computers with the MC68881 that leaves out the console functions.

## Using the Online Standard Libraries Reference

The THINK Project Manager comes with an online *Standard Libraries Reference*: a version of THINK Reference that includes a database for the THINK C standard libraries.

If you have Symantec C++, you have another database that describes the Symantec C++ class libraries IOStreams and Complex.

Note

The version of THINK Reference that comes with the THINK Project Manager does not contain the Macintosh Toolbox databases. To obtain those databases, you must buy the full version of THINK Reference, available wherever you bought THINK C or Symantec C++.

If you want a printed copy of the *Standard Libraries Reference*, use the order form that came with your package.

### Looking up a function

*For some shortcuts to looking up information in THINK Reference, see "Looking Up Definitions and Documentation" in Chapter 10.*

To look up a standard library function when you're in the THINK Project Manager, select the function name and choose **Find in THINK Reference** from the **Search** menu. You can also go to the Finder, double-click the THINK Reference icon, and type the function's name. Usually after a few seconds you'll see the THINK Reference window shown in Figure 21-1.

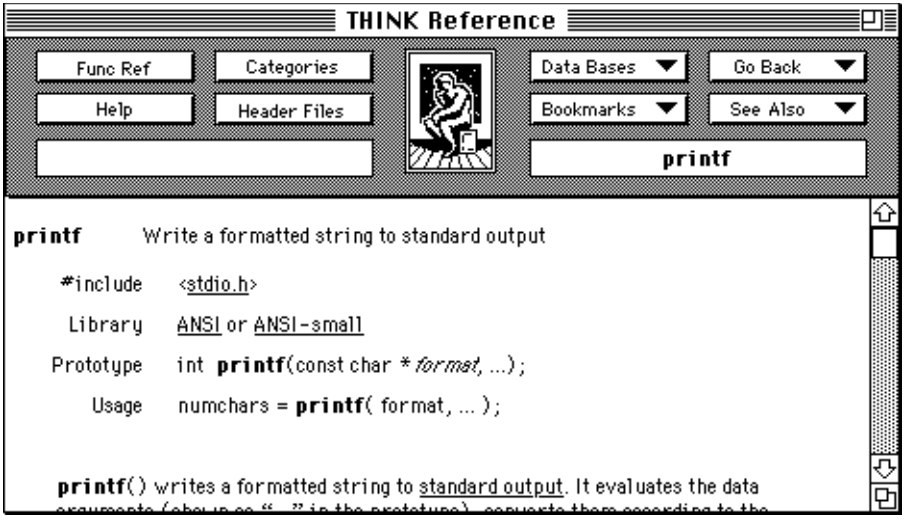


**Figure 21-1** THINK Reference window

Note

If you see the error message `Can't find THINK Reference`," look at "Installation," Chapter 2, for more information on how to install THINK Reference.

Note

If you moved the databases since you installed them, THINK Reference displays a dialog that asks you to locate them. See "Finding lost databases" later in this chapter for more information.

The top part of the THINK Reference window is the button panel, which contains buttons that help you move around the database. The box on the right side is the Page Title field and contains the name of the current field. The bottom part is the information area, which contains a page from the database.

### Looking up a topic

If your THINK C documentation refers you to a topic in the online *Standard Libraries Reference*, start THINK Reference, type the name of the topic, and press Return. THINK Reference brings you to the page that deals with that topic.

### Moving around THINK Reference

*Sometimes THINK Reference uses underlined words for emphasis. You can differentiate these from hyperlinks because the cursor changes to a magnifying glass only over hyperlinks.*

You move around THINK Reference with hyperlinks. When you click a hyperlink, you go to a page that describes a related topic. Whenever the cursor is over a hyperlink, it changes to a magnifying glass ( ). Hyperlinks appear as **bold underlined** words or as plain underlined words. Bold underlined hyperlinks usually refer to functions, and plain underlined hyperlinks refer to types, variables, or fields of structures.

You can also go to a hyperlink by typing its name. What you type appears in the box on the left side of the button panel. If the hyperlink is on the current page, THINK Reference scrolls the page to bring it into view and selects it. If the hyperlink isn't on the current page, THINK Reference displays in the Page Title field the name of the hyperlink that matches what you've typed. Press Return to go to the hyperlink's page.

Read THINK Reference's online documentation for more information on the features discussed here and on other features such as searching, printing, and setting preferences. Just click the Help button in the button panel.

### Reading a function reference page

Most function reference pages contain the four sections in Figure 21-2: summary, description, returns, and example. Some may omit the returns and example sections.

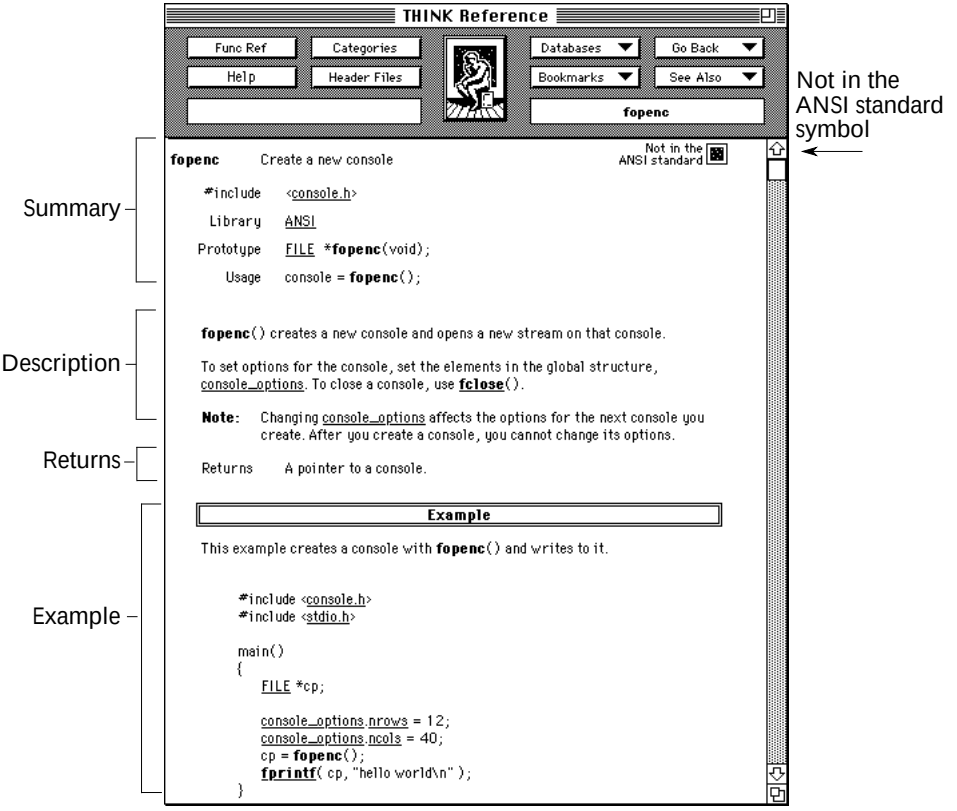


**Figure 21-2** A function reference page

The summary section briefly describes the function in five lines, as described in Table 21-3.

| This line… | Contains… |
|---|---|
| function name | The name of the function and a short description. |
| `#include` | The header file that declares the function. Include it in any source file that calls the function. |

**Table 21-3** Summary section of a function reference page

| This line… | Contains… |
|---|---|
| Library | The libraries that contain the function. Include one of these libraries in your project. |
| Prototype | The function's declaration from the header file. |
| Usage | An example of how to use the function in a statement. |

**Table 21-3** Summary section of a function reference page *(Continued)*

If the function is not part of the ANSI standard, the Not in the ANSI standard symbol appears at the right side of the summary section. If you plan to compile your program with another ANSI C compiler, avoid functions that aren't in the standard.

The description section tells you what the function does and describes each of its arguments in detail. This section may contain notes that call attention to important information.

The returns section summarizes what the function returns. If the function sets `errno` when there's an error, this section also describes `errno`'s possible values. If a function doesn't return a value, it doesn't have a returns section.

Most functions also have an example section that illustrates how to use the function in a program. To copy the example to the clipboard, choose **Copy Code Examples** from the **Edit** menu. You can then paste it into a source file and try it out.

### Using the tables of contents

The Quick-Jump buttons in the upper-left corner of the THINK Reference window bring you to tables of contents that help you look up the information you need.

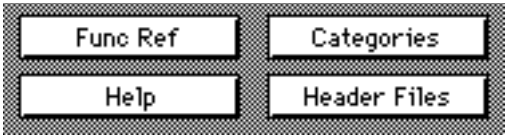


**Figure 21-3** The Quick-Jump buttons

Which button you use depends on what you're looking for and how much you know before you start looking. For instance, if you want more information on a function and you know its name starts with

`fget`, you would start in the Func Ref page. If you know that `stdio.h` has a function for printing error messages, the best place to start would be in the Header Files page.

When you're looking up something in a table of contents, remember that you can type the name of a hyperlink to select it. If the hyperlink doesn't appear in the window, THINK Reference scrolls the page until it appears. To go to the hyperlink's page, press Return.

Table 21-4 summarizes how you use the Quick-Jump buttons to look up information.

| Use this… | To look up… |
|---|---|
| Func Ref | A function, if you know part of its name |
| Categories | A function, if you know roughly what it does |
| Help | Online help |
| Header Files | A data structure, type, or function, if you know its header file |

**Table 21-4** Uses of Quick-Jump buttons

**Func Ref**

The Func Ref page is an alphabetical listing of every function in the THINK C standard libraries. To get to this page, you click the Func Ref button in the button panel, choose **Func Ref** from the **Reference** menu, or press Command-R.

**Categories**

The Categories page is a list of categories, such as I/O Functions, UNIX Functions, and Time Functions. This page is useful if you know what the function does but don't know its name. Each category has a page that contains hyperlinks to all the functions listed on it.

To get to the Categories page, you click the Categories button in the button panel, choose **Categories** from the **Reference** menu, or press Command-T.

**Help**

The Help page contains links to pages that explain how THINK Reference works. To get to the Help page, you click the Help button in the button panel, choose **Help** from the **Reference** menu, or press Command-N.

## Header Files

The Header Files page contains an alphabetical list of hyperlinks to the header files in the THINK C standard libraries. Each hyperlink takes you to a page that describes the types, macros, global variables, and functions that the header file defines.

To get to the Header Files page, you click the Header Files button in the button panel, choose **Header Files** from the **Reference** menu, or press Command-H.

### Finding lost databases

If you have moved a database since you installed it, THINK Reference asks you to locate it. THINK Reference expects to find its databases in the folder that contains the THINK Reference application, which is where the installer places them. THINK Reference displays the dialog shown in Figure 21-4. Once the name of the folder containing the database is displayed in the button under the prompt Use all database files in, click the button. THINK Reference remembers the location of the database.

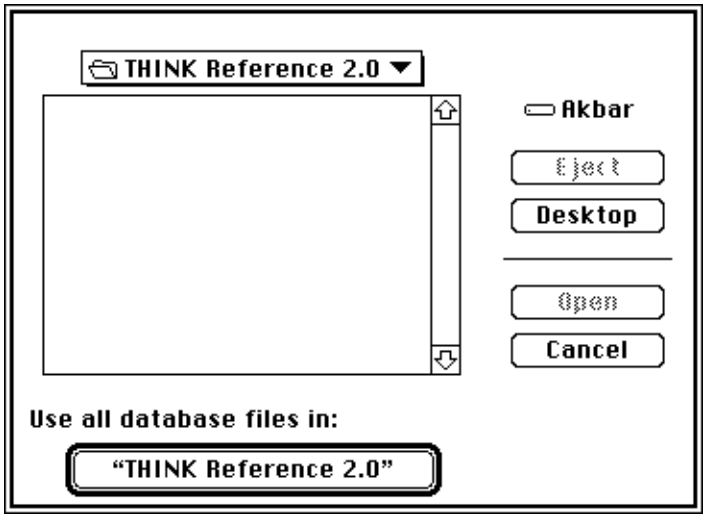


**Figure 21-4** Selecting the database folder

Whenever you move a database, THINK Reference brings up this dialog the next time you use it. If you want THINK Reference to bring up this dialog, hold down the Shift key as THINK Reference is starting up. You may need this dialog if you have more than one set of databases and want to switch between them.